Approximate Inference of Bayesian Networks through Edge Deletion

Julie Thornton
Department of Computer Science, Kansas State University
234 Nichols Hall, Manhattan, KS 66506
jas3466@cis.ksu.edu

Abstract
In this paper, we introduce two new algorithms for approximate inference of Bayesian networks that use edge deletion techniques. The first reduces a network to its maximal weight spanning tree using the Kullback-Leibler information divergence as edge weights, and then runs Pearl’s algorithm on the resulting tree for linear-time inference. The second algorithm deletes edges from the triangulated graph until the biggest clique in the triangulated graph is below a desired bound, thus placing a polynomial time bound on inference. When tested for efficiency, these two algorithms perform up to 10,000 times faster than exact techniques. See www.cis.ksu.edu/~jas3466/research.html for more information.

Introduction
We propose two new approximate inference algorithms that use edge deletion to reduce the complexity of the network and guarantee polynomial-time inference. The first algorithm, called “reduction to polytree”, computes the maximal weight spanning tree of a Bayesian network using the Kullback-Leibler (KL) information divergence as edge weights. We can then run Pearl’s polytree algorithm for linear-time inference [Pe88].

Our second new algorithm, called “bounding clique sizes”, deletes edges to force a desired bound on the clique sizes in the triangulated graph before running the inference portion of the Lauritzen-Spiegelhalter (LS) junction tree algorithm [LS88]. This clique bound forces a polynomial time bound on the inference algorithm.

Past Edge Deletion Techniques
Edge deletion techniques for approximate inference were developed by Kjaerulf in [Kj93] and van Engelen in [vE96]. Both authors used KL divergence as edge weights, and tried to delete edges with low weights (and hence little mutual information). Kjaerulf chose edges for removal from the triangulated graph of a Bayesian network, while van Engelen removed edges from the original network. van Engelen also devised an optimized computation of KL divergence requiring only local information. Both authors were able to delete the appropriate edges to stay within a desired error bound.

Description of Algorithms
This section describes the two new approximate inference algorithms, reduction to polytree and bounding clique sizes.

Reduction to Polytree
Figure 1 contains the pseudocode for the reduction to polytree (RP) algorithm. The notation B=(G=(N, A), P) for a Bayesian network B means that B is made up of a directed graph G with nodes N and arcs A, and a probability distribution P.

Bounding Clique Sizes
Figure 2 contains the pseudocode for the bounding cliques sizes (BCS) algorithm. The algorithm takes as input a Bayesian network B, an initial perfect elimination scheme f (we used the result of the maximum cardinality search [MCS] algorithm), and a desired bound b on the clique sizes in the triangulated graph. Essentially, this algorithm removes edges from the original graph G until the fill-in triangulation of G given f has a maximum clique size that is no bigger than the desired bound b.

Consider first the step that removes “unnecessary” moral edges after the selected edge (u, v) has been removed. A moral edge (u, x) is unnecessary if it was not an edge in the original graph G until the fill-in triangulation of G given f has a maximum clique size that is no bigger than the desired bound b.

Figure 1. Reduction to polytree algorithm

Figure 2. Bounding clique sizes algorithm
edge adjacent to some node in the maximum clique whose deletion renders the most moral edges unnecessary, and which does not serve as a moral edge. If no such edge exists, then we choose any non-moral edge whose deletion maximizes the number of unnecessary moral edges.

boundingCliqueSizes(B = (G = (N, A), P), f, b)
G' = (N', A') ← moralized graph of G
FillIn ← the fill-in triangulation of G' given f
maxClique ← nodes in the max clique of FillIn

while sizeof(maxClique) > b
pick an edge (u, v) ∈ A
G' ← (N', A\(\backslash\)u, v)
remove unnecessary moral edges from G'
FillIn ← the fill-in triangulation of G' given f
maxClique ← nodes in the max clique of FillIn

run LS on the junction tree for FillIn [LS88]
report the posterior probabilities for each node

Figure 2. Bounding cliques algorithm

Results

We tested the two new algorithms on six Bayesian networks: Asia, Insurance, Water, Alarm, Barley, and CPCS-179. The characteristics of the networks are in Table 1.

<table>
<thead>
<tr>
<th>Network</th>
<th>Nodes</th>
<th>Edges</th>
<th>Max Clique Size (MCS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asia</td>
<td>8</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>Insurance</td>
<td>27</td>
<td>52</td>
<td>10</td>
</tr>
<tr>
<td>Water</td>
<td>32</td>
<td>66</td>
<td>12</td>
</tr>
<tr>
<td>Alarm</td>
<td>37</td>
<td>46</td>
<td>5</td>
</tr>
<tr>
<td>Barley</td>
<td>48</td>
<td>84</td>
<td>9</td>
</tr>
<tr>
<td>CPCS-179</td>
<td>179</td>
<td>239</td>
<td>9</td>
</tr>
</tbody>
</table>

Table 1. Summary of Bayesian networks used

The total time comparison for three different algorithms -- RP, BCS (with desired clique bounds of 5, 6, and 8), and LS -- appears in Figure 3. LS is slightly faster than RP and BCS for Alarm and CPCS-179, but is much slower than RP and BCS on Water and Barley. In the case of Barley, LS takes 7.5 hours to complete, while RP finishes in 0.6% of that time, and BCS finishes in 0.008% of that time.

Table 2 shows the root mean-squared error (RMSE) for the approximate posterior probabilities found by RP and BCS. RP produces reasonable RMSE values (less than 0.1) for all networks except Alarm, and BCS has reasonable RMSE values for all networks except Water. Furthermore, while the RMSE for Water is poor for the clique bounds 5 and 6, it drops to 0.0025 with a clique bound of 8. However, Figure 3 shows that the total time for BCS with a clique bound of 8 is still 0.6% of the time for LS.

<table>
<thead>
<tr>
<th>Network</th>
<th>RP</th>
<th>BCS (b=5)</th>
<th>BCS (b=6)</th>
<th>BCS (b=8)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asia</td>
<td>0.0012</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Insurance</td>
<td>0.0540</td>
<td>0.0327</td>
<td>0.0086</td>
<td>0.0050</td>
</tr>
<tr>
<td>Water</td>
<td>0.0210</td>
<td>0.1915</td>
<td>0.1914</td>
<td>0.0025</td>
</tr>
<tr>
<td>Alarm</td>
<td>0.1900</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Barley</td>
<td>0.0164</td>
<td>0.0187</td>
<td>0.0303</td>
<td>0.0108</td>
</tr>
<tr>
<td>CPCS-179</td>
<td>0.0952</td>
<td>0.0896</td>
<td>0.0610</td>
<td>0.0004</td>
</tr>
</tbody>
</table>

Table 2. RMSE comparison between techniques

Discussion

Both RP and BCS showed promising results. They ran up to 10,000 times faster than LS on the more complex networks. Furthermore, the error introduced by these two techniques was minimal – both algorithms had RMSE values under 0.1 for five out of six networks. BCS had RMSE values below 0.1 for all six networks if the appropriate maximum clique size bound was used.

The techniques of reducing a network to a polytree or bounding the clique sizes in a junction tree are novel approaches to creating efficient approximate inference algorithms for Bayesian networks. As such, there is much room for research in constraining the structure of a network in order to speed up the inference process.

References


